`Duwak`: Dual Watermarks in Large Language Models

Chaoyi Zhu
TU Delft,
Delft, Netherlands
c.zhu-2@tudelft.nl

Jeroen Galjaard
TU Delft,
Delft, Netherlands
J.M.Galjaard@tudelft.nl

Pin-Yu Chen
IBM Research,
New York, USA
pin-yu.chen@ibm.com

Lydia Y. Chen
TU Delft,
Delft, Netherlands
lydiaychen@ieee.org

Abstract

As large language models (LLM) are increasingly used for text generation tasks, it is critical to audit their usages, govern their applications, and mitigate their potential harms. Existing watermark techniques are shown effective in embedding *single* human-imperceptible and machine-detectable patterns without significantly affecting generated text quality and semantics. However, the efficiency in detecting watermarks, i.e., the minimum number of tokens required to assert detection with significance and robustness against post-editing, is still debatable. In this paper, we propose, `Duwak`, to fundamentally enhance the efficiency and quality of watermarking by embedding dual secret patterns in both token probability distribution and sampling schemes. To mitigate expression degradation caused by biasing toward specific tokens, we design a contrastive search to watermark the *sampling scheme*, which minimizes the token repetition and enhances the diversity. We theoretically explain the interdependency of the two watermarks within `Duwak`. We evaluate `Duwak` extensively on Llama2 and Vicuna under various post-editing attacks, against four state-of-the-art watermarking techniques and combinations of them. Our results show that `Duwak` marked text achieves the highest watermarked text quality at the lowest required token count for detection, up to 70% tokens less than existing approaches, especially under post paraphrasing. Our code is available at https://github.com/chaoyitud/Dual-Watermarks.

1 Introduction

Large language models (LLMs) are widely adapted for natural language tasks, including copywriting (OpenAI, 2022), machine-translation (Zhang et al., 2023), questioning and answering (Touvron et al., 2023a), and code generation (Rozière et al., 2023). While LLMs achieve remarkable and human-like performance, there are increasing risks of abusing LLM's (Kuditipudi et al., 2023) to produce incorrect and adversarial content on social media and to commit fraud in academic rights. Watermarking LLM content is one of the essential solutions to govern the LLM applications and guardrail their misuse and harm to the society, even requested by the governmental policies (Veale and Zuiderveen Borgesius, 2021). Much like physical watermarks, embedding watermark signals on LLM-generated text provides the means to trace content to their generator as well as the LLM models that constantly evolve.

Key criteria for watermarking generative language models are multiple folds: having minimal degradation of the generated content quality, imperceptible to humans for avoiding alteration, detectable by machines for rigorous auditing, and robust against post-text editing. Recent studies show that a **single watermark** pattern can be hidden in generated text through either altering the underlying token probability distribution (Kirchenbauer et al., 2023a; Yoo et al., 2023; Fernandez et al., 2023) or modifying the sampling strategy (Kuditipudi et al., 2023; Christ et al., 2023; Aaronson, 2022). While the existing watermarks achieve multiple criteria, their practicability on short texts and post-edited text is limited, as the minimum number of tokens required for successful detection, e.g., low false positive rate, under those scenarios is high.

In this paper, we propose a dual watermarking approach, `Duwak`, which improves the watermark detection efficiency and text quality by embedding two independent secret patterns into the token probability distribution and sampling scheme. To detect the watermark, `Duwak` searches for the union of these two watermarks—the enabler for efficient detection with a low token count. Under `Duwak`, we first modify the pre-activation logits of pseudo-randomly selected tokens seeded by a function (i.e., hash) of a prior token sequence and

Findings of the Association for Computational Linguistics ACL 2024, pages 11416–11436
August 11-16, 2024 ©2024 Association for Computational Linguistics

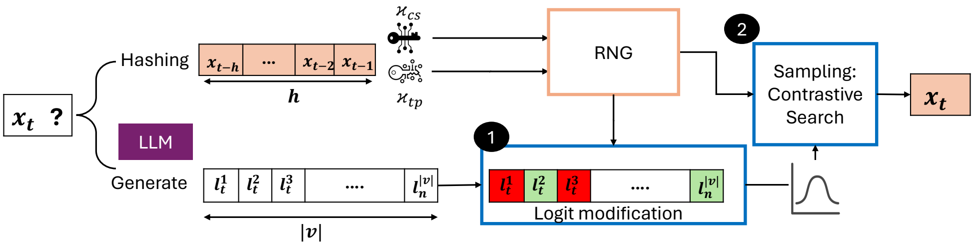


Figure 1: `Duwak`: dual watermarking LLMs. To generate a token x_t, `Duwak` embeds two secret patterns, governed by random number generation seeded by two private keys and prior tokens, via (i) pre-activation logit modification and (2) a contrastive search sampling strategy.

a secret key, similar to green-red list watermarking approaches (Kirchenbauer et al., 2023a; Yoo et al., 2023; Wang et al., 2023b; Zhao et al., 2023). Consecutively, we randomly split the token into the normal and watermark sampling set, which embeds an additional random number sequence seeded by a second secret key.

The challenge lies in efficiently detecting watermarks without degrading quality. It is known that watermarking activation signals inevitably degrade the text quality due to the bias term on a pseudo-random selection of tokens (Welleck et al., 2020; Kuditipudi et al., 2023). To counteract this degradation, we advocate the use of a quality aware sampling scheme—the contrastive search, which limits token sampling to top-k tokens resulting in the lowest similarity w.r.t. previous generated tokens. Unlike the popular cryptographic sampling, the contrastive search marks sampling patterns, thereby improving the text expression, improving the diversity of token selection and thus the watermarked text quality (Ren et al., 2023; Liu et al., 2023).

Our contributions are summarized in the following:

Improving watermark efficiency, through `Duwak`'s joint dual watermarks patterns in the token probability distribution and sampling scheme.

Increasing generation diversity and robustness, by avoiding expression degradation of watermarked text. `Duwak` includes a novel quality-aware sampling scheme based on the contrastive search.

Empirical evaluation showing the effectiveness of `Duwak` against existing watermark solutions under nine post-editing attacks. Thereby showing the minimum number of tokens required to reach detection accuracy is up to 70% lower than related work, with nearly the best text quality and diversity.

2 Background

LLM text synthesis Large language models are typically transformer-based neural networks, denoted by M and parameterized by θ. Internally, these models tokenize the vocabulary into a set, $\mathcal{V}$, and generate a token sequence indexed by i, $x_{i\geq 0}$, based on the prompt text, which is represented as a token sequence with negative index $x_{i<0}$. Generally, generative LLMs 'complete' a provided sequence (prompt) in an auto-regressive fashion, i.e., the token of t-th position is based on the prompt and thus far generated tokens, i.e., tokens $x_{i<t}$, from here on notated simplified as $x_{<t}$. The token generation consists of two stages. First, the LLM estimates the probability scores of the succeeding token x_t^n for all $|\mathcal{V}|$ tokens at the position t, $\forall n \in \mathcal{V}$ by softmaxing the model's output logits, $l_t^n = l_\theta\left(\cdot \mid x_{<t}\right)^n$,

$$p_\theta(\cdot|x_{<t})^n = \texttt{softmax}(l_t)^n, \forall n \in \mathcal{V}.$$

The second step is to sample the token based on the estimated probability distribution. Common sampling schemes differ in their objectives and complexity: greedy search, beam-search, top-k sampling (Fan et al., 2018a), nucleus-sampling (top-p) (Holtzman et al., 2020), multinomial (random) sampling, and contrastive search (Su et al., 2022).

Watermarking LLM Watermarks are typically embedded in the process of next-token generation through altering: (i) the logit and probability (Kirchenbauer et al., 2023a; Yoo et al., 2023; Lee et al., 2023a) and (ii) the sampling scheme (Aaronson, 2022; Christ et al., 2023; Kuditipudi et al., 2023). To change the probability distribution, the key idea is to split the vocabulary tokens into a green (preferred) and red list, $\mathcal{V} \in \mathcal{G} \cup \mathcal{R}$, via a random number that is hashed from a secret key and an aggregate of previous h tokens. The number of green tokens is controlled by hyper-parameter γ by taking $|\mathcal{G}| = \gamma|\mathcal{V}|$. The logit values of green tokens receive a bias δ, thereby increasing their probability estimates, thus increasing the likelihood of them being selected. The sampling scheme can remain the same as the original LLM. Consequently, watermarked text is expected to have an increase in the number of green tokens. In contrast, sampling-based approaches are deterministic while keeping the model's next token probability estimate untouched. Aaronson (2022) propose an exponential scheme and choose the token $x_t = \arg\max_{n \in \mathcal{V}} \left\{ (r^n)^{\frac{1}{p^n}} \right\}$, where p is the unaltered probability vector and $r \in [0,1]^{|\mathcal{V}|}$, is the random number vector generated by hashing the prior h tokens and the secret key.

Detecting Watermarking Detecting watermarks requires inspecting a sequence of N tokens and computing their watermark likelihood score, S_N. The exact score computation depends on watermarking methods. In the case of logit modification through the green-red list (Kirchenbauer et al., 2023a), every token is classified into the green or red list based on the random split, conditioned on the random number sequence seeded by prior tokens and secret key. The total number of green tokens is the score. As for the sampling approach, e.g., Aaronson (2022), computes a pre-determined threshold is exceeded by negative summation of $\sum_{i \in N} \ln(1 - r_i)$. Here the intuition lies in the fact that a token with low p_i would require an r_i arbitrarily close to 1, thus limiting their contribution to the computed score. This metric essentially measures the aggregate deviation from the expected distribution of tokens under the manipulation of random number vector r.

Watermarking Measures There are multiple measures for watermarking algorithms: text quality, detection efficiency, and robustness. In terms of quality, perplexity (Kirchenbauer et al., 2023a; Wang et al., 2023a; Kuditipudi et al., 2023) metrics, rating from another (larger) LLM (Kocmi and Federmann, 2023; Piet et al., 2023), and diversity (Su et al., 2022; Kirchenbauer et al., 2023b) are used to assess the (watermarked) LLM text. As for detection efficiency and robustness, it measures the number of N tokens needed to achieve significant detection tests under differentattacks, e.g., insertion, deletion, and paraphrasing (Piet et al., 2023).

Z-statistic and p-value (Kirchenbauer et al., 2023a) are commonly used to evaluate the significance of the detection test, assuming the detection scores follow the normal distribution with a mean of μ and standard deviation of σ. The null hypothesis of the detection test is that H_0: the text is unwatermarked. The Z-statistics represents the normalized observed score value, which is subtracted by the estimated mean and standard deviation. And, its corresponding p-value represents the probability of having a normalized score higher than observed Z under the H_0, i.e., the text is not watermarked.

3 `Duwak`: Dual Watermarking

The objective of `Duwak` is to maintain the watermarked text quality while keeping high detection efficiency, i.e., high detection confidence by inspecting a low number of tokens. `Duwak` embeds two secret watermark signals sequentially in the token probability distribution and token sampling scheme as shown in Fig. 1. To mitigate the text distortion caused by modifying the token probability, we design a contrastive search sampling scheme that increases the diversity via selecting tokens with low similarity among the top-k ones. We elucidate the interdependency through the joint watermarking scheme of `Duwak`, demonstrating that the two watermarks can be integrated efficiently with an efficiency guarantee.

3.1 Token Probability Watermark

To generate token x_t from a given prompt and prior generated token sequence, `Duwak` first alters the token probability distribution $p_t \to \hat{p}_t$ by altering the logit values for a subset of $n \in \mathcal{V}$. Specifically, a secret key κ_{tp} and the prior sequence of window h, i.e., $x_{t-h \le t \le t-1}$, are inputs to a pseudo-random number generator, `RNG`, for generating a fixed-length pseudo-random number sequence. Consecutively, each random number is used to split the token into binary types, i.e., green v.s. red. Generally, the secret keys used during watermarking are only known to the owner. Such a design guarantees that only the watermark owner can identify and decode the watermarked tokens, embedding a layer of security and specificity within the generated text. Following (Kirchenbauer et al., 2023a), a bias term, δ, is added to the logit of tokens on the favored list, termed green list, while keeping logits of non-biased tokens, coined red list, remains unchanged. As the token probability distribution is computed as taking the softmax function on the logit, shown in Eq. 1, the token probability distribution is thus modified, risking text quality degradation. The higher the δ value, the higher the distortion to the probability and thus higher the possibilityof degradation in text quality. We note that `Duwak` is compatible with any probability modification proposed in existing watermarking algorithms, and we, in practice, adopt the algorithms derived in (Kirchenbauer et al., 2023b). More specifically, defining p_t^n as,

$$\hat{p}_t^n = \frac{\exp\left(l^n + \mathbb{1}\left[n \in \mathcal{G}\right]\delta\right)}{\sum_{i \in V} \exp\left(l^i + \mathbb{1}\left[i \in \mathcal{G}\right]\delta\right)}, \quad (1)$$

where $\mathbb{1}[c]$ is 1 when c holds, otherwise 0.

3.2 Contrastive Search Watermark

One of the known limitations of LLM is anisotropic representation—repetitive wording and degenerated expression (Ethayarajh, 2019; Su et al., 2022; Su and Collier, 2023). To avoid such degradation, (Su and Collier, 2023) define a self-similarity measure of token x_t with respect to all other tokens in the vocabulary $\mathcal{V}$, i.e., $x_{j \in V \setminus \{i\}}$. A higher value of self-similarity suggests a more isotropic representation space. To address the isotropic degradation, the token is then sampled to maximize the summation of the weighted token probability and the penalty of self-similarity.

We adapt such a contrastive search principle into a watermark sampling scheme in a sliding window manner. This approach not only incorporates a distinctive sampling scheme but also significantly enhances the diversity of text generation. Effectively reducing token repetition and mitigating text degeneration, leading to more coherent and varied output. Here, token at position t, are split into two sets, (i) $\mathcal{C}$ with a probability η, subject to contrastive search sampling, and (ii) $\overline{\mathcal{C}}$ with a probability $1-\eta$, where standard multinomial sampling is applied. The segmentation into $\mathcal{C}$ and $\overline{\mathcal{C}}$ is facilitated by a pseudo-random number generator that leverages a hashing value of previous tokens and a watermark key, κ_{cs}.

Contrastive searching sampling aims to reduce the similarity to the prior L token sequence. For all the contrastive set, we limit the selection to the top-k tokens, i.e., $V_t^{(k)}$, with the highest kth probability. The top-k sampling is designed to reduce the risk that unlikely tokens are sampled (Fan et al., 2018b), reducing the search space of contrastive search. We then choose a token, $v \in V_t^{(k)}$ that maximizes the weighted probability and minimizes self-similarity with respect to the prior L tokens.

We first define the similarity between x_t and $x_{t-L\leq j<t}$ as the cosine distance between their hidden state, $s\left(h_{x_t}, h_{x_j}\right) = \cos(h_{x_i}, h_{x_j})$, where h_{x_i} and h_{x_j} represent the last layer hidden states in the model of token x_i and x_j respectively, and cos is the cosine-similarity between embeddings. Extending it to the L window, the self-similarity of x_t is computed as the maximum value with respect to all L prior tokens, $x_{t-L\leq j<t}$, i.e., $s_L(x_t) = \max_{t-L\leq j<t}\left\{s\left(h_{x_t}, h_{x_j}\right)\right\}$.

A sliding window L increases generation efficiency by limiting the similarity computation to L preceding tokens. Moreover, it increases robustness against attacks by limiting the context on which the watermark is conditioned. The token is finally chosen by maximizing the weighted probability, $\hat{p}_t^v$ and similarity penalty, $\cdot s_L(x_t^v)$, where α is a hyper-parameter that balances the importance of the weighted probability of the token against its self-similarity penalty.

$$x_t = \underset{v \in \mathcal{V}^{(k)}}{\arg\max}\left\{(1-\alpha)\cdot \hat{p}_t^v - \alpha \cdot s_L(x_t^v)\right\} \quad (2)$$

Algorithm 1 `Duwak` Token Generation.

Input: $\theta, \kappa_{\text{tp}}, \kappa_{\text{cs}}$
Params: $\texttt{RNG}, k, L, \texttt{Hash}, \eta, s_L$
Output: $x_t \in \mathcal{V}$

function DUWAKGENERATE
  $seed \leftarrow \texttt{hash}(x_{<t})$
  $r \leftarrow \texttt{RNG}(seed, \kappa_{\text{cs}})$
  procedure TOKENPROBWATERMARK
    $\mathcal{G} \leftarrow \texttt{RNG}(seed, \kappa_{\text{tp}})$
    Compute $\hat{p}_t^n$ as Eq. 1
  procedure CSWATERMARK
    if $r < \eta$ **then**
      $\mathcal{V}_t^{(k)} \leftarrow \text{top}_k(\hat{p}_t)$
      Contrastive search as Eq. 2
    else
      $x_t \sim \text{Multinomial}(\hat{p}_t)$
  return x_t

3.3 Detection in Duwak

To detect the watermarks within a text sequence x of length T, we employ hypothesis testing to differentiate between the null-hypothesis $\mathcal{H}_0$: "the text is generated naturally" and the alternative hypothesis $\mathcal{H}_1$: "the text is generated with `Duwak`."

Given the incorporation of two distinct watermarks, we treat the detection of each as two

Algorithm 2 `Duwak` Watermark Detection.

Input: $\theta, \kappa_{\text{tp}}, \kappa_{\text{cs}}$
Params: γ, T, η, M, L
Output: p-value $\in [0, 1)$

function DuwakDetection
  procedure Compute_P_{tp}
    $\phi_{tp} \leftarrow \sum_{t=1}^{T} \mathbb{1}\left[x_t \in \mathcal{G}_t\right]$
    $z_{tp} = \frac{\phi_{tp} - \gamma T}{\sqrt{T\gamma(1-\gamma)}}$
    $P_{tp} = 1 - \Phi(z_{tp})$
  procedure Compute_P_{cs}
    $P_{cs} \leftarrow 1$
    for $\kappa_m \in \{\kappa_m \mid \kappa_m \neq \kappa_{cs}\}_{i=m}^{M}$ **do**
      $P_{cs} \leftarrow P_{cs} + \mathbb{1}\left[\phi_{cs}^{(\kappa_m)} \geq \phi_{cs}^{(\kappa_{cs})}\right]$
    $P_{cs} \leftarrow \frac{1}{1+M} P_{cs}$
  $P \leftarrow 1 - F_{\chi^2}(x, 4)$ where $x = -2(\ln(P_{tp}) + \ln(P_{cs}))$
  return P

separate and independent tests. We first detect token probability and constrastive search watermark independently and compute their p-values, namely, P_{tp} and P_{cs}, against the full hypothesis that the text is not altered by token probability (constrastive search) watermark. We then apply Fisher's method (Fisher, 1922) to that combining p-values from these two independent tests into a single statistic follows a chi-square (χ^2) distribution with $d = 4$ degrees of freedom:

$$-2(\ln(P_{tp}) + \ln(P_{cs})) \sim \chi^2(4).$$

Furthermore, the resulting p-value P, derived from the chi-square distribution, is given as:

$$P = 1 - F_{\chi^2}\left(-2\left(\ln(P_{kgw}) + \ln(P_{cs})\right), 4\right),$$

where F_{χ^2} is the cumulative distribution function (cdf) for the chi-square distribution. This provides a unified statistical measure to assess the presence of watermarks in the text.

To compute the p-values for both watermarks, we resort to a concept of score, ϕ, which represents the discernible discrepancy between watermarked and non-watermarked texts. Higher the score, stronger the evidence of watermarked text. We explain how to derive the p-values from their detection scores.

P-value of token probability watermark (P_{tp}). We use the number of detected green-listed tokens of the T token sequence as the score, i.e., $\phi_{tp} = \sum_{t=1}^{T} \mathbb{1}\left[x_t \in \mathcal{G}_t\right]$, where $\mathcal{G}_t$ is generated from `RNG(hash`$(x_{<t})$`,`κ`)`, which based on the watermark key and preceding tokens. To assert its significance, we apply a Z-test on $z_{tp} = \frac{\phi_{tp} - \gamma T}{\sqrt{T\gamma(1-\gamma)}}$ and then compute the corresponding p-value, as $P_{tp} = 1 - \Phi(z_{tp})$, where Φ is the cumulative distribution function of normal distribution.

P-value of contrastive search watermark (P_{cs}). As the score distribution in non-watermarked text is unknown, our proposed score for the contrastive search watermark is based on self-similarity difference between the contrastive set, $\mathcal{C}$ and non-contrastive set $\overline{\mathcal{C}}$, split by using the key κ. Intuitively, the score is higher when the correct key, κ_{cs}, is used to split the set, compared to using arbitrary keys. To assert the statistical significance in the score difference, we propose to compare the scores between using the known private key κ_{cs} and other M randomly chosen keys, $\kappa_{1 \leq m \leq M}$.

We first formally define these two sets as, $\mathcal{C}$ and $\overline{\mathcal{C}}$. Following that we define the score of contrastive search watermark using any key κ as

$$\phi_{cs}^{(\kappa)} = -\left(\frac{\sum_{t \in \mathcal{C}} s_L(x_t)}{|\mathcal{C}|} - \frac{\sum_{t \in \overline{\mathcal{C}}} s_L(x_t)}{T - |\mathcal{C}|}\right). \tag{3}$$

We then compute the score for the key, κ_{cs} and κ_m, and count the number of times that the score of using κ_m is higher than κ_{cs}, Finally, we approximate the p-value of contrastive search as,

$$P_{cs} = \frac{1}{M+1}\left(1 + \sum_{m=1}^{M} \mathbb{1}\left[\phi_{cs}^{(\kappa_{tp})} \geq \phi_{cs}^{(\kappa_{cs})}\right]\right).$$

3.4 Theoretical Analysis

The following theorem shows that two watermarks do not influence each other.

Theorem 3.1 (Green List Tokens using topk)**.** *Given $\mathcal{X} = \{x_1, \ldots, x_T\}$ from an LLM with green list fraction γ, and token n's adjusted probability at t follows Eq. 1. Define $V_t^{(k)}$ as the set of top-k tokens by $\hat{p}_t^n$,with $x_t \sim$ Uniform$(V_t^{(k)})$. Given $\mathbb{E}|V_t^k|_{\mathcal{G}} \geq \nu$, then the expectation and variance of the count of green list tokens $|x|_{\mathcal{G}}$ in $\mathcal{X}$ are bounded as follows:*

$$\mathbb{E}|x|_{\mathcal{G}} \geq \frac{\nu}{k} T, \ \mathrm{Var}\,|x|_{\mathcal{G}} \leq T \cdot \nu\,(k - \nu)\,k^{-2}.$$

In our theorem, we describe a bound that elucidates the interdependency between two watermarks. We model the contrastive search as akin to uniformly sampling from the top-k candidates. By influencing the selection among the top-k tokens based on historical similarity, with a large vocabulary size $|\mathcal{V}|$ and a small k, the process effectively approximates random selection. The theorem sets bounds on the expectation and variance of "green list" tokens, based on the limit of mean green token selection within the top-k candidates. This effectively outlines the interdependency between the two watermarks in our Duwak.

4 Evaluation

In this section, we first detail the evaluation setup on the LLM prompts and evaluation tasks. The evaluation metrics are the quality of watermarked text and the token count needed to achieve certain detection p values under normal conditions and various post-editing attacks. We compare Duwak against existing single watermark techniques and combinations thereof.

4.1 Evaluation setup

Prompt. For evaluation, we use open-ended generation (Su and Collier, 2023) and MarkMyWords' (Piet et al., 2023) structured tasks. The detailed settings can be found in Appendix C.1.

Models. In our experiments, we utilize two primary models: Llama2-7b (Touvron et al., 2023b) and Vicuna-7b-v1.5 (Zheng et al., 2024).

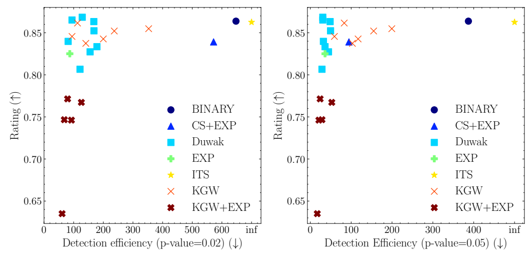
Evaluation metrics. To evaluate watermark methods, we use the following metrics: Diversity, MAUVE, Rating, and Detection efficiency. Details on these metrics and their configurations are provided in Appendix C.2.

Baseline. A summarized baseline overview is given in Tab. 1 (i) the Kirchenbauer-Geiping-Wen (KGW) algorithm (Kirchenbauer et al., 2023a), Exponential (EXP) (Aaronson, 2022), Binary (Christ et al., 2023) (BINARY), Inverse Transform Sampling (Kuditipudi et al., 2023) (ITS) and Contrastive Search (CS) (ours) are the single watermarking algorithm, and (ii) KGW-EXP, CS-EXP, and Duwak (ours) are the dual watermark algorithms. We highlight where the watermark signals are inserted in the token probability or sampling. For dual watermarking schemes, we conduct the χ^2 test on the p-value of each watermark as Duwak.

Hyper-parameter setting. For a fair comparison across algorithms, we limit the hashing input to the first preceding token to generate watermark seeds for all watermarking algorithms. As for the fraction of green tokens, $\gamma|\mathcal{V}|$ under KGW probability modification, we use a fixed $\gamma = 0.5$. The detection window of Duwak is set as $L = 50$ token, and the probability of contrastive search is $\eta = 0.5$.

Comp. \ Alg.	KGW	EXP	ITS	BINARY	KGW-EXP	EXP-CS	Duwak
$\Delta P(x_t\|x_{<t})$	KGW	-	-	-	KGW	-	KGW
Sampler	Multi	Exp	Inverse	Binary	Exp	CS	CS

Table 1: Watermarking algorithms: token probability modification, and sampling scheme. '-' denotes no token probability distribution modification.



(a) $p = 0.02$ (b) $p = 0.05$

Figure 2: Rating v.s. token efficiency under different watermarking methods and hyper-parameter settings for different detection p-values.

4.2 Results

Quality v.s. detection efficiency. We summarize the overall results in Tab. 2 and Tab. 3, highlighting the difference among human, unaltered LLM, and watermarked LLM text from all the watermarking methods. First of all, human-written text shows the highest diversity and MAUVE scores. Regarding the quality of the watermarked text, Duwak ranks as the first or the second-best method in terms of diversity, MAUVE, and rating, achieving similar results as the unaltered LLM text. CS achieves the highest diversity and MAUVE as expected among the single watermarks. Among dual watermarks, the direct combination of the common probability modification (KGW) and token sampling (EXP) deteriorates text quality due to the EXP sampling method, which heavily biases the modified token probability. Overall, including contrastive search improves the text quality to its CS-less counterpart.

The efficiency of detection of watermarks measures the number of tokens needed to detect watermarks with p-values of 0.02. EXP-CS is the only exception because both watermarks are embedded in the sampling process and interfere with each other, arguing the risk of blending multiple watermarks. On the other hand, a single watermark

Watermark	Human	No Watermark	KGW	EXP	BINARY	ITS	CS	KGW-EXP	EXP-CS	Duwak
Diversity (%) (↑)	93.62	86.66	81.41	39.58	44.56	78.72	**86.53**	17.90	83.83	83.98
MAUVE (%) (↑)	100.0	82.36	75.5	55.87	55.57	79.02	<u>80.71</u>	27.03	77.58	**82.18**
Rating (%) (↑)	-	87.28	86.15	82.56	**87.10**	86.25	83.74	77.14	83.91	<u>86.51</u>
Dection efficiency (↓)	-	-	113	<u>89.5</u>	847	>1024	>1024	**79.5**	572	94.5

Table 2: Comparison of watermarking methods on different metrics on Llama2-7b. Arrows point to the direction of better performance: a downward arrow (↓) means lower is better, and an upward arrow (↑) means higher is better. **Bold**/<u>underlined</u> text means the best/second-best score.

Watermark	No Watermark	KGW	EXP	BINARY	ITS	Duwak
Rating (%) (↑)	84.1	82.1	82.0	82.2	**83.4**	<u>83.1</u>
Detection efficiency (↓)	-	101.5	**71**	252	>1024	<u>82.5</u>

Table 3: Comparison of watermarking methods on different metrics on Vicuna-7b-v1.5.

requires a significantly higher number of tokens, especially for BINARY, ITS, and CS, strengthen the watermarked text's robustness and quality.

Fig. 2 provides a sensitivity perspective of watermark methods under different hyper-parameter settings and p-values, 0.02 and 0.05. Specifically, different δ values are used in KGW probability modification. Duwak shows more consistent performance across all δ's compared to KGW, i.e., slightly higher rating and lower tokens with a lower variance. This trend continues for a p-value of 0.05, with a more pronounced difference in their required token counts. Specifically, when compared to the best KGW watermark, our algorithm requires $\sim$ 40 fewer tokens. When p-values are smaller, the number of tokens needed for detection increases considerably.

Duwak achieves the best quality efficiency ratio, high diversity, MAUVE, and rating, using fewer tokens to detect watermarks accurately compared to other watermarking methods.

Post-editing attack robustness. Here, we evaluate the robustness of Duwak under different post-editing attacks, i.e., attacks that alter the tokenization. Specifically, we consider contraction, lowercase, misspelling, repetition, swap, synonym, translation, typo, and paraphrase attacks from MarkMyWords (Piet et al., 2023). Tab. 4 and Tab. 5 present the efficiency of reaching a p-value of 0.02 under KGW, EXP, and Duwak. Such a selection is based on the observation in Tab. 2 that only these three methods achieve reasonable text quality while inspecting roughly 100 tokens.

In Tab. 4, while EXP shows the best efficiency in the no-attack scenario (through significant inference quality), Duwak requires significantly lower

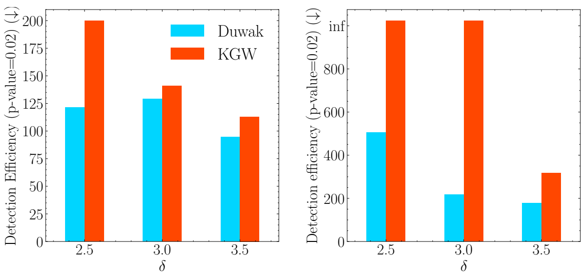
Attack	Conf.	EXP	KGW	Duwak
None		**89.5**	113	94.5
Contraction		88.5	114	**87.5**
Lowercase		**106**	146	113
Repetition&deletion		**83.5**	108	87.0
Paraphrase	GPT3.5	238	322	**193**
Misspelling	25%	93.5	119	**82.5**
	50%	148	147	**114**
Swap	5%	83.0	113	**77.5**
	10%	83.0	113	**82.0**
Synonym	25%	90.5	118	**81.0**
	50%	100	134	**100**
	75%	126	169	**112**
	100%	170	213	**125**
Translation	FR	118	147	**114**
	RU	156	195	**148**
TypoAttack	5%	221	221	**177**
	10%	389	337	**301**

Table 4: Attacked detection efficiency on Llama2-7b, lower is **better**.

Attack	Conf.	EXP	KGW	Duwak
None		**71**	101.5	82.5
Contraction		**72.5**	99	87.5
Lowercase		**108**	130	113.5
Repetition&deletion		**72.5**	114	89
Paraphrase	GPT3.5	1024	582	**328**
Misspelling	25%	124	128.5	**116.5**
	50%	**82**	96	86
Swap	5%	84.5	96	**84**
	10%	**84**	100.5	101.5
Synonym	25%	**80.5**	118	91
	50%	**97**	131	116
	75%	142.5	139.5	**126**
	100%	206.5	156	**126.5**
Translation	FR	**102**	155	106.5
	RU	**137.5**	168	148
TypoAttack	5%	212	209.5	**185**
	10%	1024	1024	**316**

Table 5: Attacked detection efficiency on Vicuna-7b-v1.5, lower is **better**.

tokens for inspection in the presence of attacks, i.e., ranging between 6 to 70%. The presence of attacks clearly increases the need to consider more tokens for all watermark methods. Let's zoom into the performance of Duwak against each of those attacks, in contrast to the cast of no attack. TypoAttack significantly increases the detection difficulty and results in a more than 3× increase in the number of tokens. Misspelling and repetition&deletion, swap, and synonym (25%) are simple attacks, even reducing the number of inspection tokens. Paragraphs and TypoAttack are where Duwak has the best performance, compared to EXP, the second-best policy. We attribute this difference to the two watermarks and no interference among them. Additionally, Duwak benefits from incorporating two distinct watermarks that operate without mutual interference, thereby enhancing its robustness. In results from Vicuna-7b-v1.5, as shown in Tab. 5, we observe similar trends in performance. However, under some attacks, particularly simpler ones, EXP achieves better efficiency. Nevertheless, in more severe scenarios, especially with strong attacks like the paraphrase attack, Duwak significantly outperforms EXP, demonstrating its robustness in handling more complex attacks.



(a) No attack. (b) Paraphrase attack.

Figure 3: Detection efficiency (↓) of Duwak and KGW with equal hyper-config under varying δ.

Impact of contrastive search sampling. Here, we highlight the impact of contrastive search compared to the single KGW watermark. In Fig. 3, we show the rating and the number of inspected tokens to achieve a p-value of 0.02 under the different distribution shifting($\delta \in \{2.5, 3, 3.5\}$) with clean and paraphrase attack versions. Unsurprisingly, Duwak outperforms KGW due to the addition of contrastive search, such the advantage diminishes with increasing δ. When δ is large, e.g., 3.5, it introduces a large distortion in the generation probability, leaving little room for Duwak to further improve the quality. In the case of the challenging paraphrasing post-attacks, shown in Fig. 4b, one can observe the clear advantage of using contrastive search. This observation again verifies our design of dual watermark, which is inherently more robust to the post-editing when compared to the token-level approaches of prior art.

5 Related Studies

Prior single watermark solutions embed the watermark signal at the token level with a modification of the generation process by modifying either the token probability distribution (Lee et al., 2023b; Wu et al., 2023; Takezawa et al., 2023) or sampling scheme (Aaronson, 2022; Christ et al., 2023; Kuditipudi et al., 2023).

Watermark in token probability distribution. Kirchenbauer et al. (2023a) design the very first single-bit watermark method for LLM text generation, splitting tokens into a green and red list using a cryptographic key.

To further improve the text quality and robustness, subsequent studies modify the criteria of green-red splits. Zhao et al. (2023) prove that global red-green splits improve robustness against post-editing attacks, whereas Kirchenbauer et al. (2023b) propose to use the minimum hashed token to determine the red-green list.

Furthermore, to improve the governance of watermarks and provide additional information, e.g., copyright and timestamp, multi-bit watermarks (Wang et al., 2023b; Yoo et al., 2023; Fernandez et al., 2023) are proposed, introducing message-specific red-green lists. We note that such watermarks split the text into multiple sections, each of which has only a single watermark in their token probability, whereas our solution embeds up to two watermarks into a single token.

Watermark in Sampling Binary watermark (Christ et al., 2023) samples the token based on the comparison of the predicted probability and the pseudo-random presentation. Because of the fixed length of pseudo-random numbers, the LLM can end up generating the same text for the same prompt. Kuditipudi et al. (2023) propose the usage of longer pseudo-random number sequences than the generated text itself and randomly choose the insertion location in the text to add the watermark. Hou et al. (2023) resort to watermarking via sentence-level sampling, which iteratively performs sentence-level rejection sampling until the sampled sentence falls within the watermarked region.

We note that orthogonal to watermark detection is more general detection of *whether* text is synthesized by LLMs (Solaiman et al., 2019; Gehrmann et al., 2019; Mireshghallah et al., 2023; Mitchell et al., 2023). However, as traceability to specific models cannot be provided, these detection works are limited in their application for the governance of synthesized text.

6 Conclusion

In this paper, we propose a dual watermark scheme for LLM, `Duwak`, which embeds human imperceptible and machine detectable watermarks in token probability distribution and sampling schemes. Combining two watermarks significantly decreases the minimum number of tokens for detecting watermarks with a desirable false positive rate, especially when encountering post-editing attacks. To avoid text quality degradation due to watermarking token probabilities, we design a contrastive search sampling scheme that samples tokens with the lowest similarity. We show the effectiveness of `Duwak` by providing a theoretical lower bound on the watermarked tokens and extensive empirical evaluation. Compared against existing single watermark solutions and combinations thereof, `Duwak` provides a better watermarked text quality. This is especially highlighted in terms of diversity, and robustness against nine post-editing attacks, using up to 70% less tokens for detection.

7 Limitation

This study introduces advancements in watermarking techniques for Large Language Models (LLMs) through `Duwak`, while also recognizing certain limitations that warrant future investigation. Firstly, our approach's effectiveness is contingent on the specific characteristics of the LLMs evaluated, primarily Llama2. Consequently, the applicability of `Duwak` to different models and subsequent versions of LLMs is a subject that merits further exploration. Moreover, our evaluation was restricted to text-generation tasks. The extension of our methodology to encompass additional tasks, such as the generation of mathematical proofs or code, remains an area requiring in-depth study. Additionally, `Duwak` necessitates conducting two separate detection processes for each watermark, which results in a decrease in detection time efficiency compared to single watermark methods.

Impact Statements

With the popularity of large language models and their applications, embedding watermarks into their generated content is an essential step toward trustworthy and responsible AI technology development and deployment. Our findings of improved watermark detection performance and utility provide novel insights into the research and practice of watermarking for large language models.

Acknowledgment

This work has been partly funded by the Dutch National Science Foundation Perspectief Project, DEPMAT.

References

Scott Aaronson. 2022. My ai safety lecture for ut effective altruism.

Miranda Christ, Sam Gunn, and Or Zamir. 2023. Undetectable watermarks for language models. *arXiv preprint arXiv:2306.09194*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 55–65. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018a. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018b. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. 2023. Three bricks to consolidate watermarks for large language models.

Ronald A Fisher. 1922. On the interpretation of χ 2 from contingency tables, and the calculation of p. *Journal of the royal statistical society*, 85(1):87–94.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: statistical detection and visualization of generated text. In *Proceedings of* *the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pages 111–116. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. Semstamp: A semantic watermark with paraphrastic robustness for text generation.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A watermark for large language models. *arXiv preprint arXiv:2301.10226*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2023b. On the reliability of watermarks for large language models.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation, EAMT 2023, Tampere, Finland, 12-15 June 2023*, pages 193–203. European Association for Machine Translation.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2023a. Who wrote this code? watermarking for code generation. *CoRR*, abs/2305.15060.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2023b. Who wrote this code? watermarking for code generation.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2023. A semantic invariant robust watermark for large language models. *CoRR*, abs/2310.06356.

Fatemehsadat Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. 2023. Smaller language models are better blackbox machine-generated text detectors. *CoRR*, abs/2305.09859.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. 2023. Mark my words: Analyzing and evaluating language model watermarks. *arXiv preprint arXiv:2312.00273*.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems*, volume 34, pages 4816–4828. Curran Associates, Inc.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. 2023. A robust semantics-based watermark for large language model against paraphrasing. *CoRR*, abs/2311.08721.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton-Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code. *CoRR*, abs/2308.12950.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203.

Yixuan Su and Nigel Collier. 2023. Contrastive search is what you need for neural text generation. *Trans. Mach. Learn. Res.*, 2023.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation. In *NeurIPS*.

Yuki Takezawa, Ryoma Sato, Han Bao, Kenta Niwa, and Makoto Yamada. 2023. Necessary and sufficient watermark for large language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023a. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Michael Veale and Frederik Zuiderveen Borgesius. 2021. Demystifying the draft eu artificial intelligence act—analysing the good, the bad, and the unclear elements of the proposed approach. *Computer Law Review International*, 22(4):97–112.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. 2023a. Towards codable text watermarking for large language models. *CoRR*, abs/2307.15992.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. 2023b. Towards codable watermarking for injecting multi-bit information to llm.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. 2023. Dipmark: A stealthy, efficient and resilient watermark for large language models.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. 2023. Advancing beyond identification: Multi-bit watermark for large language models.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 41092–41110. PMLR.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable robust watermarking for ai-generated text. *CoRR*, abs/2306.17439.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

A Nomenclature

α Parameter balancing the importance of token probability and self-similarity in contrastive search.

$\boldsymbol{\theta}$ Large language model parameters used for text generation.

δ Bias term added to the logits of tokens on the green list to alter their probabilities.

η Probability determining whether contrastive search or multinomial sampling is used for token generation.

γ Portion of the vocabulary designated as the green list in the token probability modification process.

$\hat{p}_t^n$ The probability distribution over tokens after applying watermark modifications.

κ_{cs}, κ_{tp} Secret keys used for embedding watermarks in the text.

$\mathcal{G}$ A subset of tokens selected for next word generation, influenced by a watermark key.

$\mathcal{V}$ The set of all possible tokens the LLM model can generate.

Φ, F_{χ^2} Cumulative distribution functions used to calculate p-values in hypothesis testing for watermark detection.

ϕ_{cs} A score for Contrastive Search watermark.

ϕ_{tp} A score computed for token token probability watermark

H_V The representation of tokens in the model's hidden layer.

k Top-k parameter defining the number of top predictions considered in the generation process.

L Sliding window length used in contrastive search to compute token similarity.

l_t The raw outputs of the LLM model for the next token, before applying the softmax function.

P, P_{cs}, P_{tp} P-values indicating the likelihood of observing the test results under the null hypothesis.

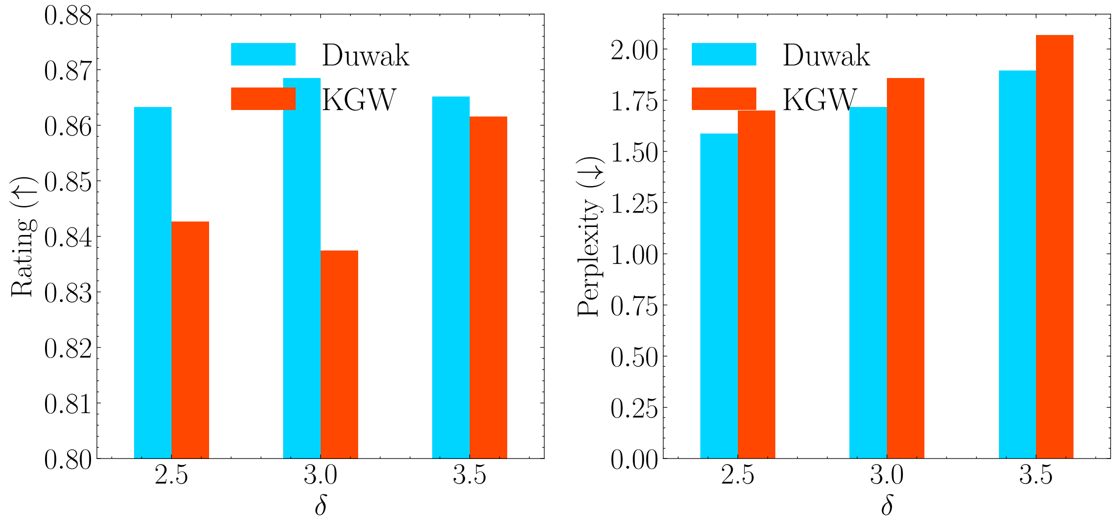
$s_L(x_t)$ A measure of a token's similarity to its preceding tokens within a sliding window of length L.

`Hash` A function used to generate a hash value based on the current context.

`RNG` A function generating pseudo-random numbers based on a seed and possibly a key.

B Additional Results

B.1 Rating and perplexity comparison between `Duwak` and KGW



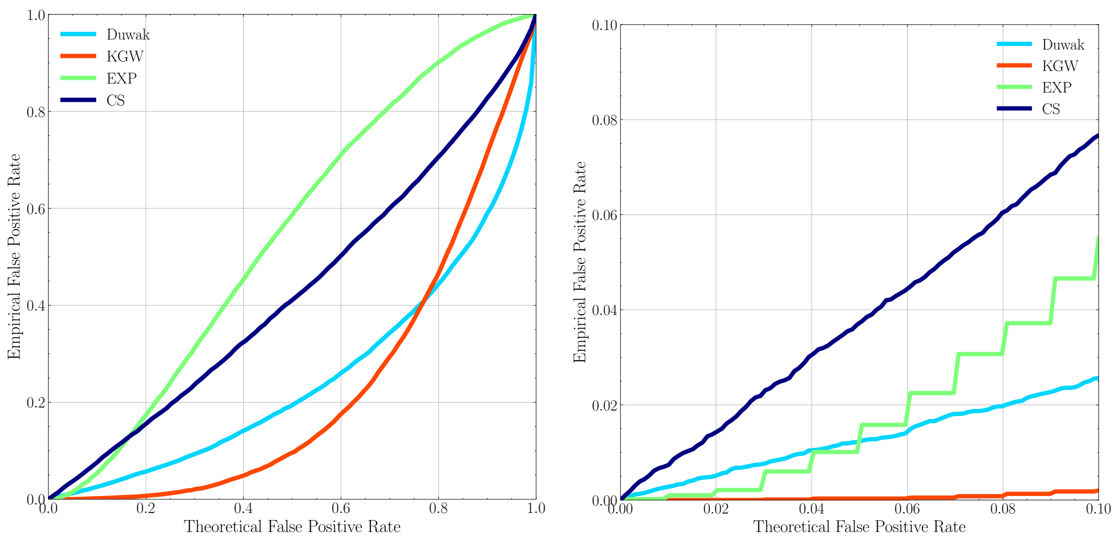
(a) No attack.

(b) Paraphrase attack.

Figure 4: Comparative analysis of `Duwak` and KGW with identical hyper-parameters under varying δ, detection efficiency (↓).

B.2 Empirical false positive rates

To assess the theoretical false positive rate (p-value) through empirical means, we utilize a the following methodology to identify unwatermarked text within the Wikitext dataset. For each watermark, we examine 10,000 samples, each with an average length of 260 tokens. The empirical false positive rate is determined by the proportion of texts erroneously identified as watermarked at the p-value threshold.



(a) Complete comparison.

(b) Zoomed-in version of Fig. 5a from 0 to 0.1.

Figure 5: Comparison of empirical false positive rate and theoretical false positive rate for different watermarks

We observe that our algorithm does not result in an empirical false positive rate (FPR) higher than the theoretical FPR. Moreover, all methods tend to overestimate the false positive rate when the p-value is lower than 0.1, particularly in the case of KGW.

B.3 Detection Efficiency Under Different p-value Thresholds

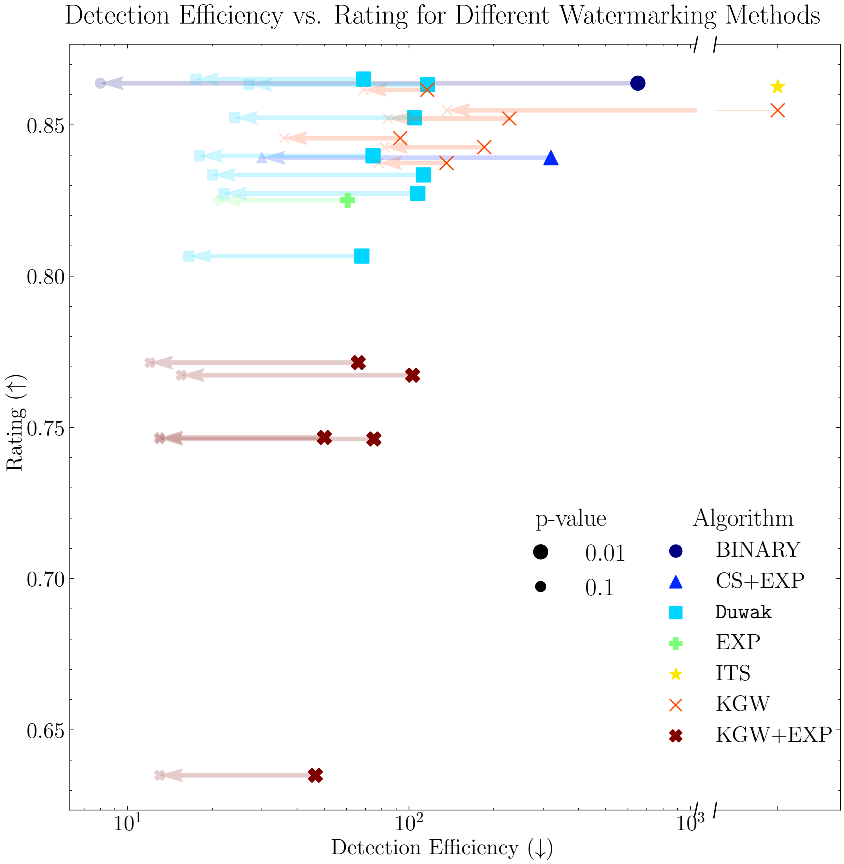


Figure 6: Detection efficiency vs. rating under different watermarking methods and hyper-parameter settings with p-value 0.01 and 0.01. Arrows are drawn between the corresponding configurations with different p-values to indicate the reduction of detection tokens required for a more lenient p-value.

B.4 Ablation Study

We conducted additional experiments on the Llama2-7b model, as shown in Table 6, to assess the effectiveness of our `Duwak` when both components are active as well as to assess the effectiveness of each individual component (KGW and CS) on the process. Table 6 summarizes the detection efficiency results, defined as the median number of minimal tokens needed for detection when the p-value reaches 0.02. These findings clearly demonstrate that the combined use of both the KGW watermark and the contrastive search watermark within our `Duwak` results in superior detection efficiency compared to the performance the individual components.

C Evaluation Setup

C.1 Prompt

For evaluation, we use open-ended generation and MarkMyWords' (Piet et al., 2023) structured tasks.

Open-ended text generation: Following Su and Collier (2023), 40 WebText corpus token prompts are used to generate completions of up to 200 tokens. Comparing the quality of watermarked generations to the datasets' human-completions indicates the watermarkings' relative effectiveness.

MarkMyWords generation tasks: Additionally, we include three tasks from the MarkMyWords dataset—book reports, story generation, and fake news generation—to mirror realistic scenarios where watermarking allows for harm mitigation and accountability.

C.2 Evaluation metrics

We use the following metrics to assess the performance and effectiveness of watermark methods.

Diversity: Accounts variance in generated content using repetition under varying n-grams ($n \in \{2, 3, 4\}$).

MAUVE: (Pillutla et al., 2021) Measures the similarity between generated and human-written text token distributions. Higher MAUVE shows better resemblance to human text.

Rating: Automated evaluation with the GPT-3.5 Turbo API, rating the relevance and fluency of watermarked texts on a 100-point scale based on zero-shot prompting.

Detection efficiency: The minimum token count required for watermark detection for a type-I error rate (p-value). Thus ensuring a precise balance between text length and the efficacy of watermark detection, highlighting our watermarking technique's efficiency.

C.3 Assessment Guidelines for Rating Responses

To systematically evaluate the text quality, each text will be graded out of a total of 100 points by GPT-3.5. The grading criteria are divided into four key categories, with points allocated as follows:

Accuracy (20 points): This measures the correctness and relevance of the response in relation to the provided prompt. Points are awarded based on how well the response adheres to factual accuracy and pertinence to the initial query or statement.

Detail (20 points): This assesses the comprehensiveness and depth of the response. A higher score is given for responses that cover the topic thoroughly, providing a nuanced understanding of the subject matter.

Grammar and Typing (30 points): This category evaluates the grammatical and typographical precision of the response. A flawless submission, free from spelling errors, punctuation mistakes, and grammatical inaccuracies, will receive full points.

Vocabulary (30 points): This criterion looks at the appropriateness and richness of the language used. Responses that utilize a diverse vocabulary appropriately and effectively to convey ideas will score higher in this category.

Points should be deducted for any deficiencies observed within each category. The total score, calculated out of 100, should be presented at the beginning of the evaluative feedback.

Prompt Template:

```
[INST] <<SYS>> You are given a prompt and a response,
and you need to grade the response out of 100 based on:
Accuracy (20 points) - correctness and relevance to the prompt;
```

Attack	Conf.	`Duwak`	KGW in `Duwak`	CS in `Duwak`
None		83	89	269
Contraction		81	87	241
Lowercase		97	114	233
Repetition&deletion		76	82	300
Misspelling	25%	84	91	543
	50%	88	99	228
Swap	5%	84	96	342
	10%	89	102	206
Synonym	25%	88.5	92	247.5
	50%	85.5	104	275.5
	75%	91	108	259.5
	100%	97.5	125	351
Translation	FR	105	153	675
	RU	107	148.5	>1024
TypoAttack	5%	85	150	517
	10%	152	225	985
Paraphrase	GPT3.5	184	312	1012

Table 6: Attacked detection efficiency on Llama2-7b for `Duwak` and it's components, lower is **better**.

```
Detail (20 points) - comprehensiveness and depth;
Grammar and Typing (30 points) - grammatical and typographical accuracy;
Vocabulary (30 points) - appropriateness and richness.
Deduct points for shortcomings in each category.
Give a total grade at the first line of the response. <</SYS>>
Prompt: {}
Response: {}
[/INST] Grade out of 100:
```

This structured approach ensures a comprehensive and fair assessment of GPT-3.5's responses, facilitating a consistent evaluation process.

D Generated text

Table 7: Completions to prompt "[INST] «SYS» You are a helpful assistant. Always answer in the most accurate way. «/SYS» Write a story about a retired astronaut that joins a risky interstellar rescue mission.. [/INST]", limited to first 3 sentences.

generator	delta	response
BINARY	1.5	John "Hawk" Hawkins had seen it all during his illustrious career as an astronaut. He had flown on dozens of missions to space, including the historic first manned voyage to Mars, and had even broken a record for the longest time spent in space. But after decades of flying through the cosmos, Hawk decided it was time to hang up his space suit and enjoy his well-deserved retirement.\n\nThat was until he received a distress call from a nearby planet that had been devastated by a catastrophic event
CS	2.5	Title: The Call of Duty\n\nRetired astronaut John "Hawk" Hawkins had seen it all during his illustrious career in space exploration. From the vastness of the cosmos to the intricate workings of the human body, he had experienced it all. But at the age of 65, he thought he had left his adventurous days behind
CS+EXP	2.5	Title: The Call of Duty\n\nJohn "Hawk" Hawkins, a retired astronaut in his early sixties, had grown accustomed to the quiet life on Earth. After decades of space travel, he had left the thrill of exploration behind and settled into a comfortable routine. However, fate had other plans.\n\nOne fateful evening, while watching the news, Hawk's eyes widened as he listened to the urgent message from Mission Control: an interstellar rescue mission was underway, and they needed his expertise
CS+KGW	2.5	Title: The Call of Duty\n\nRetired astronaut John "Jack" Jackson had spent his entire adult life working for NASA, logging over a hundred days in space and earning a reputation as one of the best pilots in the business. But after a near-fatal accident on a mission to Mars, he was forced to retire and leave his beloved space travel behind.\n\nYears went by, and Jack found himself growing restless and unfulfilled in his post-NASA life. He spent his days golfing and fishing, but something was missing
EXP	1.5	Title: The Courageous Retiree\n\nJohn "Hawk" Hawkins, a grizzled astronaut in his early sixties, had long since left his spacefaring days behind. Or so he thought. When he received the phone call from Mission Control, his heart raced with a mixture of excitement and trepidation
KGW	2.5	Title: The Call of Duty\n\nMission Briefing:\n\nMission Objective: Retrieve the last surviving member of the Icarus Expedition, stranded on a hostile exoplanet.\n\nMission Location: Planet Xylophia-IV, located 4.5 light years from Earth.\n\nMission Profile: The Icarus Expedition was sent to explore Xylophia-IV over 5 years ago. Upon reaching the planet's atmosphere, their spacecraft encountered unexpected storms and malfunctions, causing them to crash-land on the hostile surface. The surviving member of the expedition, Captain Emma Taylor, has been stranded on the planet for over 4 years, with limited supplies and medical equipment

Table 8: Completions to prompt "[INST] «SYS» You are a helpful assistant. Always answer in the most accurate way. «/SYS» Write a book report about 'A Clockwork Orange', written by Anthony Burgess. [/INST]", limited to first 3 sentences.

generator	delta	response
BINARY	-	Sure, I'd be happy to help you with a book report on "A Clockwork Orange" by Anthony Burgess!\n\nIntroduction:\n\n"A Clockwork Orange" is a dystopian novel written by Anthony Burgess in 1962. The novel is set in a future society where violence and sexuality are Conditioned into the population through a peculiar Conditioning method called the "Ludovico technique." The story follows the protagonist, Alex, a young man who is imprisoned for his vicious crimes and undergoes the Ludovico treatment to reduce his criminal urges. However, upon his release, Alex finds that he is unable to enjoy his favorite pastime – raping and Ultraviolence – due to the side effects of the treatment.\n\nPlot Summary:\n\nThe novel begins with Alex, a charismatic and violent young man, who lives in a dystopian society where Conditioning is a norm
CS	-	Title: A Clockwork Orange\nAuthor: Anthony Burgess\nPublished: 1962\nPages: 208\n\nBook Report:\n\n"A Clockwork Orange" is a dystopian novel written by anthony Burgees in 1962. The story takes place in a future society where violence and sexuality are socially accepted and encouraged. The protagonist, Alex DeLarge, is a charismatic and intelligent young hoodlum who leads a gang of teenage thugs in a world where "ultraviolence" is the norm.\n\nThe novel explores themes of free will, morality, and rehabilitation through Alex's journey
CS+EXP	-	Title: A Clockwork Orange\nAuthor: Anthony Burgess\nPublished: 1962\nPages: 208\n\nBook Report:\n\n"A ClockWORK Orange" is a dystopian novel written by Anthony Burgee in 1962. The story takes place in a future society where violence and sexual deviance are socially accepted and encouraged. The protagonist, Alex, is a charismatic and intelligent young hoodlum who narrates the events of the novels with a unique slang and dialect.\n\nThe novel begins with Alex and his droogs (friends) committing a series of violent crimes, including rape and murder
CS+KGW	2.5	Sure, I'd be happy to help you with your book report on "A Clockwork Orange" by Anthony Burgess!\n\nTitle: "A Clockwork Orange"\nAuthor: Anthoyn Burgess\nPublication Date: 1962\nGenre: Dystopian Novel, Satire, Young Adult\n\nPlot Summary:\nSet in a dystopian future, "A Clockwork Orange" follows the story of Alex, a charismatic and violent young hoodlum, who is arrested and sentenced to a controversial rehabilitation program called the "Ludovico technique." This program is designed to condition Alex to detest violence and sex, and to promote a sense of empathy and compassion towards others. However, the program has unintended consequences, and Alex becomes increasingly isolated and disillusioned with society.\n\nThroughout the novel, Burgess explores themes of free will, morality, and the nature of evil. Burgess challenges the reader to question the morality of punishing individuals for their actions, and to consider the ethics of manipulating people's emotions and behaviors through conditioning.\n\nCharacters:\n\n* Alex: The protagonist of the novel, Alex is a violent and charismatic young hoodlum who is the narrator of the story

EXP	1.5	Title: A Clockwork Orange\nAuthor: Anthony Burgess\nPublication Date: 1962\nGenre: Dystopian Novel, Horror, Science Fiction\n\nBook Report:\n\nSet in a dystopian near-future, Anthony Burgess's "A Clockwork Orange" follows the story of Alex, a charismatic young hoodlum who, along with his gang, commits gruesome acts of violence and rape. The novel explores themes of free will, morality, and the impact of societal conditioning on human behavior.\n\nTitle Analysis:\nThe title "A Clockwork Orange" is a clever play on words, referencing both the novel's dystopian setting and the lead character's name. The term "clockwork orange" is a contradictory phrase, as oranges are typically associated with nature and growth, while clockwork implies mechanical, predetermined movement
KGW	2.5	Title: A Clockwork Orange\nAuthor: Anthony Burgess\nPublished: 1962\nPage Count: 307\nGenre: Dystopian Fiction, Satire, Novel\n\nSummary:\n\nA Clockwork Orange is a dystopian novel written by Anthony Burgess, published in 1962. The story takes place in a futuristic society where violence and sexuality are condoned and encouraged. The protagonist, Alex DeLarge, is a charismatic young hoodlum who is sentenced to a form of experimental conditioning called the "Ludovico technique" in an effort to reform him

A Proofs for Section 3 (Duwak: Dual Watermarking)

Theorem 3.1 (Green List Tokens using topk)**.** *Given $\mathcal{X} = \{x_1, \ldots, x_T\}$ from an LLM with green list fraction γ, and token n's adjusted probability at t follows Eq. 1. Define $V_t^{(k)}$ as the set of top-k tokens by $\hat{p}_t^n$,with $x_t \sim$ Uniform$(V_t^{(k)})$. Given $\mathbb{E}|V_t^k|_{\mathcal{G}} \geq \nu$, then the expectation and variance of the count of green list tokens $|x|_{\mathcal{G}}$ in $\mathcal{X}$ are bounded as follows:*

$$\mathbb{E}|x|_{\mathcal{G}} \geq \frac{\nu}{k}T,\ \mathrm{Var}\,|x|_{\mathcal{G}} \leq T \cdot \nu\,(k-\nu)\,k^{-2}.$$

Proof. **Expectation:** The expected number of green list tokens, $\mathbb{E}|x|_{\mathcal{G}}$, is calculated as the sum of expectations over all tokens being selected from the green list across all T steps. Given that x_t is uniformly chosen from the set of top-k tokens $V_t^{(k)}$, the probability of choosing a green list token at any step t is the fraction of green list tokens in $V_t^{(k)}$, which is $\frac{|V_t^k|_{\mathcal{G}}}{k}$. Therefore:

$$\mathbb{E}|x|_{\mathcal{G}} = \sum_{t=1}^{T} \frac{|V_t^k|_{\mathcal{G}}}{k} = \frac{1}{k}\sum_{t=1}^{T} \mathbb{E}\left[\left|V_t^k\right|_{\mathcal{G}}\right].$$

Given $\mathbb{E}\left|V_t^k\right|_{\mathcal{G}} \geq \nu$, it follows that:

$$\mathbb{E}|x|_{\mathcal{G}} \geq \frac{\nu}{k}T.$$

Variance: For the variance, considering the sum of independent but not identically distributed Bernoulli trials, each trial's success probability is the fraction of green list tokens at step t, $\frac{|V_t^k|_{\mathcal{G}}}{k}$. The variance of a Bernoulli variable with probability p is $p(1-p)$. Therefore, the variance of the total count of green list tokens is:

$$\mathrm{Var}\,|x|_{\mathcal{G}} = \sum_{t=1}^{T} \frac{|V_t^k|_{\mathcal{G}}}{k}\left(1 - \frac{|V_t^k|_{\mathcal{G}}}{k}\right).$$

This simplifies to:

$$\mathrm{Var}\,|x|_{\mathcal{G}} = T \cdot \mathbb{E}\left[\frac{|V_t^k|_{\mathcal{G}}}{k}\left(1 - \frac{|V_t^k|_{\mathcal{G}}}{k}\right)\right].$$

By applying Jensen's Inequality, due to the concavity of the function $f(x) = x(1-x)$ for x in [0,1], and given $\mathbb{E}\left|V_t^k\right|_{\mathcal{G}} \geq \nu$, we derive:

$$\mathrm{Var}\,|x|_{\mathcal{G}} \leq T \cdot \frac{\nu}{k}\left(1 - \frac{\nu}{k}\right).$$

□

Theorem A.1 (Green List Tokens in Duwak)**.** *Given $\mathcal{X} = \{x_1, \ldots, x_T\}$ from an LLM with green list fraction γ, and adjusted probability of token n at t being*

$$\hat{p}_t^n = \frac{\exp(l^n + \mathbb{1}[n \in \mathcal{G}]\delta)}{\sum_{i\in V}\exp(l^i + \mathbb{1}[i \in \mathcal{G}]\delta)},$$

define $V_t^{(k)}$ as top-k tokens by $\hat{p}_t^n$, with $x_t \sim$ Uniform$(V_t^{(k)})$. The expectation and variance of green list tokens, $|x|_{\mathcal{G}}$, are bounded by:

$$\mathbb{E}|x|_{\mathcal{G}} \geq AT,$$
$$\mathrm{Var}\,|x|_{\mathcal{G}} \leq AT(1-A)(k+T-1)k^{-1},$$

with

$$A = \frac{\gamma\beta S^\star}{1+(\beta-1)\gamma}$$

under an approximation of top-k sampling to a stochastic sampling based on adjusted probability without replacement for k times. So (x_t) from $V_t^{(k)}$ approximates a stochastic process without replacement, reflecting the dynamic adjustment of probabilities as tokens are selected.

Proof. Referencing Lemma E.1 from Kirchenbauer et al. (2023a), when a token index v is sampled from the watermarked distribution, the probability that the token is from the green list in the top-k candidates is

$$\mathbb{P}[v \in \mathcal{G}] \geq \frac{\gamma\beta}{1+(\beta-1)\gamma} S_t.$$

For simplification in the proof, we rewrite $\left|V_t^k\right|_{\mathcal{G}}$ as V_t^k and $|x|_{\mathcal{G}}$ as x.

Then, the expectation of the number of green list tokens in the top-k can be expressed as:

$$\mathbb{E}_k[V_t^k] = k \cdot \frac{\gamma\beta}{1+(\beta-1)\gamma} S^t.$$

Given V_t^k, the expectation of the number of green list tokens in $\mathcal{X}$ is

$$\mathbb{E}_t[x \mid V_t^k] = \frac{T}{k}\mathbb{E}_k[V_t^k].$$

Therefore, the total expectation of green list tokens in $\mathcal{X}$ becomes

$$\mathbb{E}_t[x] = \mathbb{E}_t\left[\frac{T}{k}\mathbb{E}_k[V_t^k]\right] = \frac{T}{k} \cdot k \cdot \frac{\gamma\beta}{1+(\beta-1)\gamma}\mathbb{E}_t S^t \geq T \cdot \frac{\gamma\beta S^\star}{1+(\beta-1)\gamma},$$

i.e.,

$$\mathbb{E}_t[X_G] \geq TA.$$

Then, the variance of the green list tokens in the top-k, considering a Bernoulli distribution:

$$\operatorname{Var} V_t^k = k\frac{\gamma\beta S_t}{1+(\beta-1)\gamma}\left(1 - \frac{\gamma\beta S_t}{1+(\beta-1)\gamma}\right).$$

Consider the variance within the top-k candidates, the variance of the green list token number in the whole sequence is

$$\operatorname{Var} x = \mathbb{E}_t[\operatorname{Var}[x \mid V_t^k]] + \operatorname{Var}[\mathbb{E}_k[x \mid V_t^k]].$$

$$\operatorname{Var}[x \mid V_t^k] = T \cdot \frac{V_t^k}{k}(1 - \frac{V_t^k}{k}),$$

$$\mathbb{E}_t[\operatorname{Var}[x \mid V_t^k]] = T\left(\frac{\mathbb{E}_t[V_t^k]}{k} - \frac{\mathbb{E}_t[(V_t^k)^2]}{k^2}\right),$$

given $\mathbb{E}_t[(V_t^k)^2] = (\mathbb{E}_t[V_t^k])^2 + \operatorname{Var}_t V_t^k$ and $\mathbb{E}_t[V_t^k] \geq kA$, $\operatorname{Var}_t V_t^k \leq kA(1-A)$,

$$\operatorname{Var}[\mathbb{E}_k[x \mid V_t^k]] = T^2 \cdot \frac{\operatorname{Var} V_t^k}{k^2} \leq T^2 \cdot \frac{kA(1-A)}{k^2},$$

combine all terms together:

$$\operatorname{Var} x \leq AT(1-A)(k+T-1)k^{-1}.$$

□